

TOPOLOGICAL APPROACH TO SOLVE P VERSUS NP

KOBAYASHI, KOJI

1. OVERVIEW

This paper talks about difference between P and NP by using topological space that mean resolution principle. I focus restrictions of antecedent and consequent in resolution, and show the influence of restrictions to computation complexity.

First, I introduce RCNF that mean topology of resolution principle to formula of CNF. RCNF is HornCNF that mean the resolution principle of CNF. Variable values of RCNF formulas are presence of restrictions of CNF formula clauses.

Second, I show the restrictions of inference rule of resolution principle. Resolution have restriction that antecedents adjoin each other. Therefore, one resolution cannot infer many restrictions, but consequent of resolution become single clause. As a result, RCNF become HornCNF. And we can make antecedents product as consequent by using some resolutions which have same joint variable. But we must combine some independent antecedents to make such products.

Third, I show the P-Completeness of RCNF. By using unit resolution, we can reduce HornCNF to RCNF with logarithm space. And RCNF is HornCNF. Therefore, RCNF is P-Complete.

Last, I show the size of RCNF that reduce CNF. We can construct CNF that some truth value assignment which make one clause false (or that formula true) become totally separated and simply independent. And we can divide the totally separated and simply independent truth value assignment over polynomial scale. Therefore, RCNF formulas which change from these formulas exceed polynomial size. And CNF is not P.

2. PREPARATION

In this paper, I use description like this.

Definition 1. About $F \in CNF$, truth value assignment that F become true is $[F]$, F become false is $\overline{[F]}$, only clauses $c \in F$ become false is $\widehat{[c]}$. For simplification, I treat $\widehat{[c]} = [F]$. Number of variables of c is $|c|$, size of F is $|F|$, number of truth value assignments of $[F]$, $\overline{[F]}$ and $\widehat{[c]}$ is $|[F]|$, $|\overline{[F]}|$ and $|\widehat{[c]}|$. The composition of the clauses $c \in F$ may be denoted by a subscript. That is, $c_{i...j...} = (x_i \vee \cdots \overline{x_j} \vee \cdots)$. The subscript of a capital letter shall be either positive or negative of a variable. For examples, $c_I, c_{\overline{I}}$ means $c_I, c_{\overline{I}} \in \{c_i, \overline{c_i}\}$, $c_I \neq c_{\overline{I}}$. Truth value assignment is v , and I treat like c .

About resolution, I will use the term “Joint Variable” as variables that positive and negative variable which are included in each antecedents and not included in consequent, and “Mixed Variables” as variables that positive and negative variable which are included in each consequent, and “Positive Antecedent” as antecedent that have positive joint variable, “Negative Antecedent” as antecedent that have

negative joint variable. We treat some resolution that have same joint variable. Such case, positive antecedent, negative antecedent and consequent become set of clauses.

For simplification, we allow the resolution like $c_{i...} \wedge c_{i\bar{j}...} \rightarrow c_{i...j...}$ by using $c_{i\bar{j}...} = \top$. I also use DNF. DNF's clause is $d_{i...}$.

3. RCNF

I introduce topology of deduction system to formula. For simplification, I treat topology as formula.

Definition 2. About $F \in CNF$, I will use the term “DCNF(Deduction CNF)” as formula that variables value are presence of restrictions of CNF formula clauses. Especially, I will use the term “RCNF(Resolution CNF)” and “ $RCNF(F)$ ” as DCNF that deduction system is resolution principle. That is, if restriction of $c_{ip...} \wedge c_{iq...}$ is true in F , $RCNF(F)$ include the resolution formula $(c_{p...q...} \vee \overline{c_{ip...}} \vee \overline{c_{iq...}})$ and $c_{ip...} = c_{iq...} = \top$. And furthermore, RCNF does not include variable that correspond to empty clause because to synchronize satisfiability of F and $RCNF(F)$.

4. RESOLUTION

Resolution have completeness. If and only if F is unsatisfiable, resolution derives empty clause. Therefore, RCNF also keep SAT. From view of the computational complexity, The following behavior is important.

Theorem 3. *Antecedents of a resolution do not become false at same truth value assignment.*

Proof. We assume that some antecedent become false at same truth value assignment. Therefore, common variables in antecedent become false at same truth value assignment. As a result, in order to carry out all antecedent true, two variables may have to be restrained.

But one clauses cannot restrain more than two variables, therefore we cannot make one clause that keep antecedent true. As a result, we must make more than two consequent and the assumption contradict with resolution condition.

Therefore, we can say that some antecedents of a resolution do not become false at same truth value assignment. \square

Theorem 4. *In a resolution, antecedents have only one joint variable. That is, antecedent and truth value assignments that antecedents become false must adjoin each other.*

Proof. We assume that some resolution have 0 or over 2 joint variable.

If resolution have 0 joint variable, antecedents of the resolution can become false at same truth value assignment. But this is contradictory mentioned above 3.

If resolution have over 2 joint variables, there is no resolution because $c_{IJp...} \vee c_{\bar{I}\bar{J}q...} \rightarrow c_{Jp...\bar{J}q...} = \top$.

Therefore, we can say that antecedents have only one joint variable. \square

Theorem 5. *In a resolution, consequent is combination of positive antecedent and negative antecedent. Especially, if positive antecedent and negative antecedent do not include same variables, consequent become product of positive antecedent and negative antecedent.*

Proof. It is obvious, therefore I omit it. If resolution have $c_{ip...}, c_{iq...}, \dots$ and $c_{ir...}, c_{is...}, \dots$ as antecedents, consequents becomes product like $c_{p...r...}, c_{p...s...}, \dots, c_{q...r...}, c_{q...s...}, \dots$. \square

5. P-COMPLETENESS OF RCNF

RCNF is P-Complete.

Theorem 6. *RCNF is P-Complete.*

Proof. Clearly RCNF is HornCNF and $RCNF \in P$, I should show that we can reduce HornCNF to RCNF in logarithm space.

To reduce HornCNF to RCNF, I show 2-step procedures.

First, I reduce HornCNF to at most 3 variables clauses HornCNF. We can reduce by using same way to reduce CNF to 3CNF. That is, each clauses change follows with new variables.

$$c_{I\bar{j}kl...} \rightarrow c_{I\bar{j}0} \wedge c_{0\bar{k}1} \wedge c_{1\bar{l}2} \wedge \dots$$

We can execute this reduction with logarithm space, pointer to consequent, pointer to variable, counter that show already used variables.

Second, I reduce this HornCNF to RCNF. We can reduce by adding resolution formula for each clauses. We can reduce HornCNF with unit resolution, therefore it is enough to keep SAT by using resolution formula that variables of antecedent decreases. That is;

$$\begin{aligned} c_{I\bar{j}k} &\rightarrow (c_{I\bar{j}k}) \wedge (c_{I\bar{k}} \vee \overline{c_{I\bar{j}k}} \vee \overline{c_j}) \wedge (c_{I\bar{j}} \vee \overline{c_{I\bar{j}k}} \vee \overline{c_k}) \wedge (c_I \vee \overline{c_{I\bar{j}}} \vee \overline{c_j}) \wedge (c_I \vee \overline{c_{I\bar{k}}} \vee \overline{c_k}) \\ c_{P\bar{q}} &\rightarrow (c_{P\bar{q}}) \wedge (c_P \vee \overline{c_{P\bar{q}}} \vee \overline{c_q}) \\ c_R &\rightarrow (c_R) \end{aligned}$$

We can execute this reduction with logarithm space, pointer to consequent, pointer to variable.

Above two reduction, we can reduce HornCNF to RCNF. Both reductions use only logarithm space, we can execute all reduction in logarithm space.

Therefor, RCNF is P-Complete. \square

6. RCNF vs CNF

Next, we consider the size of RCNF which reduce CNF. I clarify the conditions which the resolution of CNF.

Definition 7. I will use the term “Truth value assignment Product” and “Clauses Product” following.

$$\begin{aligned} (v_{P...}, v_{\bar{P}...}) \times (v_{Q...}, v_{\bar{Q}...}) &= \{v_{P...Q...}, v_{\bar{P}...Q...}, v_{P...Q...}, v_{\bar{P}...Q...}\} \\ (c_{P...} \wedge c_{\bar{P}...}) \times (c_{Q...} \wedge c_{\bar{Q}...}) &= c_{P...Q...} \wedge c_{\bar{P}...Q...} \wedge c_{P...Q...} \wedge c_{\bar{P}...Q...} \end{aligned}$$

Using products, I define special CNF like this;

Definition 8. In $F \in \overline{CNFSAT}$, I will use the term “Fusion CNF” and “FCNF” as set of F that have $[\hat{c}] : c \in F$ that is totally separated and simply independent. That is,

$$\begin{aligned} \forall F \in FCNF &\left(\exists c \in F \left(v_{PQ...} \in [\hat{c}] \rightarrow v_{\bar{P}Q...} \notin [\hat{c}] \right) \right) \\ \forall F \in FCNF &\left(\exists c \in F \left(v_{S...U...}, v_{S...V...}, v_{T...U...} \in [\hat{c}] \rightarrow v_{T...V...} \notin [\hat{c}] \right) \right) \end{aligned}$$

In other words,

$$\begin{aligned} \forall F \in FCNF \left(\exists c \in F \left(\forall v_P, v_Q \dots \left((v_P, v_{\overline{P}}) \times (v_Q \dots) \notin [\widehat{c}] \right) \right) \right) \\ \forall F \in FCNF \left(\exists c \in F \left(\forall v_S \dots, v_T \dots, v_U \dots, v_V \dots \left((v_S \dots, v_T \dots) \times (v_U \dots, v_V \dots) \notin [\widehat{c}] \right) \right) \right) \end{aligned}$$

Theorem 9. *In $F \in CNF$, $RCNF(F)$ have antecedent that have all variables if $[\widehat{c}]$ is totally separated and simply independent.*

Proof. We assume that $[\widehat{c}]$ is totally separated and simply independent and $RCNF(F)$ does not have antecedent that have all variables of F .

From assumptions, F have $c_{\overline{J}} \dots$ that is false at any $[\overline{c}] \setminus [\widehat{c}]$. Mentioned above 4, antecedent must adjoin each other. Therefore, if we make resolution with c and $c_{\overline{J}} \dots$, we must divide $[\overline{c}] \setminus [\overline{c_{\overline{J}} \dots}]$ and $[\overline{c}] \cap [\overline{c_{\overline{J}} \dots}]$, and must adjoin $[\overline{c_{\overline{J}} \dots}]$ and $[\overline{c}] \setminus [\overline{c_{\overline{J}} \dots}]$ as antecedent. And we must make all combination of each clauses in F . Therefore, we must use clauses that correspond to $[\widehat{c}]$ as antecedent.

I define $RNC(f)$ to correspond to $[\widehat{c}]$. Therefore, we can resolute clauses that correspond to only $[\widehat{c}]$ from f . From assumptions, f is not included clauses that correspond to $[\widehat{c}]$ itself. $[\widehat{c}]$ is disconnected, therefore f have disconnected structure such as all clauses have 0 or over 2 joint variable. Mentioned above 4, resolution have only one joint variable. Therefore, f have joint variable that does not include F .

We think to make $c_{K \dots} \in [\widehat{c}]$ by using resolution of f . We can think 2 case, a) $c_{K \dots}$ have antecedent that shares other resolution, or b) $c_{K \dots}$ does not have antecedent that shares other resolution.

First, we think a). Consequent from the resolution of same joint variable is different each other. Consequent is the product of positive antecedents and negative antecedents. But assumptions that $[\widehat{c}]$ is simply independent contradict that $[\widehat{c}]$ is product of positive and negative antecedents.

Second, we think b). In this case, antecedents of the resolution of $c_{K \dots}$ do not connect other antecedents of resolution of $[\widehat{c}] \ni c_{L \dots} \neq c_{K \dots}$. All joint variable is different each other and f is totally separated, antecedents of $c_{K \dots}$ and antecedents of $c_{L \dots}$ is disconnected. Therefore these antecedents cannot make resolution because of mentioned above 4. Therefore, resolution of $c_{K \dots}$ and $c_{L \dots}$ in f have no common resolution and we cannot treat together. That is, f have each equivalent formula of $c_{K \dots}$ and $c_{L \dots}$. But assumptions that f does not have clauses that include all variables of F .

Therefore, we can say from the reductio ad absurdum that F have antecedent that have all variables. \square

Next, I clarify size of FCNF. For simplification, I treat truth value assignment as binary digits.

Definition 10. I will use the term “ECNF(Expand CNF)” as $E = ECNF(F) \in CNF$ like that;

$$[E] = \left\{ vw \mid v \in [F], \quad w = 2^{2k} - 1 - v, \quad k \in N, \quad 0 \leq k \leq \frac{|\langle F \rangle|}{2} + 1, \quad v < w \right\}$$

Theorem 11. $F \in FCNF \rightarrow ECNF(F) \in FCNF$

Proof. I clarify that $[ECNF(F)]$ is totally separated and simply independent.

First, I clarify that $[ECNF(F)]$ is totally separated. Mentioned above condition 10, each w of $vw \in [ECNF(F)]$ have more than two different digits. Therefore

all vw of same v is totally separated. On the other hand, all v is totally separated because of the condition of $F \in FCNF$. Therefore all vw are totally separated.

Second, I clarify that $[ECNF(F)]$ is simply independent. I use proof by contradiction. I assume that $ECNF(F)$ is not simply independent. Therefore, $(p, q) \subset [F]$, $p \neq q$ and $(p, q) \times (r, s) \subset [ECNF(F)]$, $r \neq s$. Mentioned above 10,

$$\begin{aligned} p + r &= 2^{2t} - 1, & p + s &= 2^{2u} - 1 \\ q + r &= 2^{2v} - 1, & q + s &= 2^{2w} - 1 \end{aligned}$$

Therefore,

$$\begin{aligned} (p + r) - (q + r) &= p - q = (2^{2t} - 1) - (2^{2v} - 1) = 2^{2t} - 2^{2v} \\ (p + s) - (q + s) &= p - q = (2^{2u} - 1) - (2^{2w} - 1) = 2^{2u} - 2^{2w} \end{aligned}$$

and $t, u, v, w \in N$, $p \neq q$, then $t = u, v = w$.

However,

$$\begin{aligned} (p + r) - (p + s) &= r - s = (2^{2t} - 1) - (2^{2u} - 1) = 0 \\ (q + r) - (q + s) &= r - s = (2^{2v} - 1) - (2^{2w} - 1) = 0 \end{aligned}$$

It become $r = s$ and contradict with the assumption $r \neq s$.

Therefore, we can say that $[ECNF(F)]$ is simply independent.

From the above, $F \in FCNF \rightarrow ECNF(F) \in FCNF$. □

Theorem 12. $R(F) = \frac{|[F]|}{|F|^m}$ $m : \text{constant}$, there exist $F \in FCNF$ like $O\left(\frac{R(ECNF(F))}{R(F)}\right) > O(1)$.

Proof. First, I construct $ECNF(F)$.

$ECNF(F)$ is the fomula that add some variables and clauses like $ECNF(F) = F \wedge L \wedge C$. C is the comparator CNF that means $v \leq w$ of $v \in F$, $vw \in ECNF(F)$. L is CNF that means $w = 2^{2k} - 1 - v$. Details of L are as follows;

$$L = l_0 \wedge l_1 \wedge \dots$$

$$\begin{aligned} l_i &= ((x_{2i} = y_{2i}) \vee (x_{2i+1} \neq y_{2i+1})) \wedge ((x_{2i} \neq y_{2i}) \vee (x_{2i+1} = y_{2i+1})) \\ &\wedge ((x_{2i} = y_{2i}) \vee (x_{2i+1} = y_{2i+1}) \vee (x_{2i+2} \neq y_{2i+2}) \vee (x_{2i+3} \neq y_{2i+3})) \end{aligned}$$

$$v = (x_0, x_1, \dots), \quad w = (y_0, y_1, \dots)$$

$$(x = y) \leftrightarrow (x \vee \bar{y}) \wedge (\bar{x} \vee y), (x \neq y) \leftrightarrow (x \vee y) \wedge (\bar{x} \vee \bar{y})$$

Second, I clarify $|ECNF(F)|$. $|L \wedge C|$ is directly proportional to $|\langle F \rangle|$. Therefore,

$$|ECNF(F)| = |F| + d \times |\langle F \rangle|$$

d is constant that means the size of CNF that is necessary to construct L, C of each digits.

Third, I clarify $|[ECNF(F)]|$. Each w correspond to v increase each two digits of w and reduce by half because of $v < w$. Therefore,

$$|[ECNF(F)]| = |[F]| \times \frac{|\langle F \rangle|}{2} \times \frac{1}{2} = \frac{|[F]| \times |\langle F \rangle|}{4}$$

I define $E_0 \in CNF$ that all $v \in [E_0]$ have only one true value like $[E_0] = \{0 \dots 001, 0 \dots 010, \dots, 1 \dots 000\}$. $[E_0]$ is totally separated and simply independent, therefore $E_0 \in FCNF$. $|\langle E_0 \rangle|$ have no limit, therefore $|E_0|, |[E_0]|$ also have no limit. I define $E_{t+1} = ECNF(E_t)$ which apply $ECNF$ recursively.

Apply $|ECNF(F)|$ to E_t . Using $|\langle E_{t+1} \rangle| = 2 \times |\langle E_t \rangle|$

$$|E_{t+2}| = |E_{t+1}| + d \times 2 \times |\langle E_t \rangle|$$

$$2 \times |E_{t+1}| = 2 \times |E_t| + 2 \times d \times |\langle E_t \rangle|$$

$$|E_{t+2}| - 3 \times |E_{t+1}| + 2 \times |E_t| = 0$$

Therefore, $|E_{t+1}| = |E_t|$ or $|E_{t+1}| = 2 \times |E_t|$. $|E_{t+1}| = |E_t|$ is case of $|L \wedge C| = \top$ and exclude it, therefore

$$\frac{|E_{t+1}|}{|E_t|} = 2 \rightarrow \left(\frac{|E_{t+1}|}{|E_t|} \right)^m = 2^m$$

Apply $|ECNF(F)|$ to E_t .

$$|E_{t+1}| = \frac{|[E_t]| \times |\langle E_t \rangle|}{4}$$

$$\frac{|[E_{t+1}]|}{|[E_t]|} = \frac{|\langle E_t \rangle|}{4}$$

From the above,

$$\frac{R(E_{t+1})}{R(E_t)} = \frac{|[E_{t+1}]|}{|[E_{t+1}]|^m} \times \frac{|E_t|^m}{|[E_t]|} = \frac{|[E_{t+1}]|}{|[E_t]|} \times \left(\frac{|E_t|}{|[E_{t+1}]|} \right)^m = \frac{|\langle E_t \rangle|}{2^{m+2}}$$

Because m is constant and $|\langle E_t \rangle|$ have no limit, then

$$O\left(\frac{R(E_{t+1})}{R(E_t)}\right) = O\left(\frac{|\langle E_t \rangle|}{2^{m+2}}\right) > O(1) \quad (as \quad t \gg 0) \quad \square$$

Theorem 13. *there exists $F \in CNF$ such that $O(RCNF(F)) > O(|F|^m)$ m : constant*

Proof. Mentioned above 9, $F \in RCNF$ have resolution that include all truth value assignments that totally separated and simply independent \widehat{c} . And Mentioned above 13, $O\left(\frac{R(E_{t+1})}{R(E_t)}\right) > O(1)$ and $|[E_t]|$ exceed polynomial size of $|E_t|$. Therefore, to reduce CNF to RCNF, we cannot include within polynomial size. \square

Theorem 14. $CNF \not\leq_p RCNF \equiv_L HornCNF$

Proof. Mentioned above 6, RCNF is P-Complete. But mentioned above ??, we cannot reduce CNF to RCNF in polynomial size. Therefore, $CNF \not\leq_p RCNF \equiv_L HornCNF$. \square

REFERENCES

- [1] Michael Sipser, (translation) OHTA Kazuo, TANAKA Keisuke, ABE Masayuki, UEDA Hiroki, FUJIOKA Atsushi, WATANABE Osamu, Introduction to the Theory of COMPUTATION Second Edition, 2008
- [2] HAGIYA Masami, NISHIZAKI Shinya, Mechanism of Logic and Calculation, 2007